

เริ่มต้นภาษา ไพทอน



จากพื้นฐานสู่การเขียนโปรแกรมในโลกยุคใหม่



กรณพงษ์ วัลย์ลักษณ์

หนังสือ เริ่มต้นภาษาไพทอน
จากพื้นฐานสู่การเขียนโปรแกรมในโลกยุคใหม่

กรณ์พงษ์ วิสัยทัศน์

โรงเรียนสาธิตแห่งมหาวิทยาลัยเกษตรศาสตร์
วิทยาเขตกำแพงแสน ศูนย์วิจัยและพัฒนาการศึกษา

คณะศึกษาศาสตร์และพัฒนศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน

หนังสือ เริ่มต้นภาษาไพทอน จากพื้นฐานสู่การเขียนโปรแกรมในโลกยุคใหม่

พิมพ์ครั้งที่ 1

เดือน ตุลาคม 2568

ราคา 149 บาท

ISBN (e-book): 978-616-629-320-3

บรรณาธิการ

นายกรณ์พงษ์ วิสัยลักษณ์

สงวนลิขสิทธิ์ตาม พ.ร.บ. ลิขสิทธิ์ โดยผู้เขียน นายกรณ์พงษ์ วิสัยลักษณ์ พ.ศ. 2568 การผลิต การคัดลอกและการลอกเลียนเนื้อหาไม่ว่าส่วนหนึ่งส่วนใดในหนังสือเล่มนี้ไม่ว่ารูปแบบใดๆ ทั้งสิ้น ต้องได้รับอนุญาตเป็นลายลักษณ์อักษรจากผู้เขียน

ผู้เขียน

นายกรณ์พงษ์ วิสัยลักษณ์

ออกแบบปกและเนื้อหา

นางพัฒนิกา วิสัยลักษณ์

จัดทำโดย

นายกรณ์พงษ์ วิสัยลักษณ์

ที่อยู่ โรงเรียนสาธิตแห่งมหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน ศูนย์วิจัยและพัฒนาการศึกษา
เลขที่ 1 หมู่ 6 ตำบลกำแพงแสน อำเภอกำแพงแสน จังหวัดนครปฐม 73140.

เบอร์โทรศัพท์ 08-1821-8685

Email: seksan.v@ku.th

Line ID: 0818218685

หนังสือ "เริ่มต้นภาษาไพทอน: จากพื้นฐานสู่การเขียนโปรแกรมในโลกยุคใหม่" เล่มนี้จัดทำขึ้นเพื่อเป็นแหล่งเรียนรู้สำหรับผู้เริ่มต้นศึกษาการเขียนโปรแกรมด้วยภาษาไพทอน (Python) ซึ่งกำลังได้รับความนิยมอย่างสูงในวงการภาษาโปรแกรมมิ่งทั่วโลก ด้วยจุดเด่นของภาษาไพทอนที่มีโครงสร้างไม่ซับซ้อน อ่านง่าย เขียนง่าย และมีไลบรารีให้ใช้งานอย่างมากมาย ทำให้เหมาะสมอย่างยิ่งสำหรับผู้เริ่มต้น และยังสามารถต่อยอดสู่การพัฒนาโปรแกรมในระดับสูงได้

เนื้อหาในหนังสือเล่มนี้ครอบคลุมตั้งแต่ประวัติความเป็นมาของภาษาไพทอน การติดตั้งเครื่องมือในการเขียนโปรแกรม ไปจนถึงหลักการพื้นฐานต่างๆ อาทิเช่น ตัวแปร ตัวดำเนินการ คำสั่งเงื่อนไข คำสั่งวนซ้ำ ฟังก์ชัน ลิสต์และการสร้างกราฟิกเบื้องต้นด้วย Turtle Graphics โดยในแต่ละบทจะมีตัวอย่างโค้ดที่อธิบายอย่างละเอียด พร้อมกับแบบฝึกหัดท้ายบทสำหรับทดสอบความเข้าใจให้กับผู้อ่าน

ผู้เขียนหวังเป็นอย่างยิ่งว่าหนังสือเล่มนี้จะเป็นประโยชน์สำหรับนักเรียน นักศึกษา ครู อาจารย์ และผู้ที่สนใจทั่วไป ในการเรียนรู้การเขียนโปรแกรมภาษาไพทอนได้อย่างมีประสิทธิภาพ และสามารถนำความรู้ที่ได้ไปต่อยอดในการพัฒนาโปรแกรม หรือศึกษาในระดับที่สูงขึ้นต่อไป

หากมีข้อผิดพลาดประการใด ผู้เขียนขออภัยและยินดีรับฟังข้อเสนอแนะเพื่อการปรับปรุงในโอกาสต่อไป

กรณ์พงษ์ วิสัยลักษณ์

ตุลาคม 2568



จุดประสงค์การเรียนรู้

เป็นเป้าหมายของแต่ละบทภายในหนังสือเล่มนี้ เพื่อให้ผู้อ่านได้เกิดการเรียนรู้ มีความรู้ความเข้าใจ และเกิดทักษะในการปฏิบัติจากหนังสือเล่มนี้

ตัวอย่างจุดประสงค์การเรียนรู้

บทที่ 1 รู้จักภาษาไพทอน



จุดประสงค์การเรียนรู้

1. มีความรู้ ความเข้าใจภาษาไพทอน
2. สามารถบอกถึงเครื่องมือในการเขียนภาษาไพทอน
3. สามารถเขียนโปรแกรมแสดงผลในภาษาไพทอนได้



เกร็ดความรู้

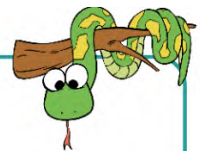
ความรู้เพิ่มเติมนอกจากที่มีอยู่ในหนังสือ เพื่อให้ผู้อ่านได้เรียนรู้มากขึ้น หรือขยายความรู้ให้กว้างมากขึ้น จากที่มีอยู่ในเนื้อหาของหนังสือ

ตัวอย่างเกร็ดความรู้



เกร็ดความรู้

Python อ่านว่า ไพทอน ซึ่งในภาษาอังกฤษหากเราแปลเป็นไทยคือ “งูหลาม”





สแกนคิวอาร์โค้ด

สแกนคิวอาร์โค้ดตัวอย่างของหนังสือ เพื่อให้ผู้อ่านได้เห็นตัวอย่างจริง ที่มีในแต่ละบทเรียน เพื่อประกอบความเข้าใจในการเรียนรู้ นอกจากอ่านในหนังสือเพียงอย่างเดียว

ตัวอย่างสแกนคิวอาร์โค้ด

ตัวอย่างที่ 1.1 คำสั่ง print แสดงข้อความที่ใช้เครื่องหมายอัฒประกาศคู่ (“ ”)



กิจกรรมฝึกทักษะ

กิจกรรมฝึกทักษะในบทเรียนเป็นการฝึกให้ผู้อ่านได้ลงมือปฏิบัติกิจกรรม และเป็นการทบทวนความรู้ที่ได้ศึกษามาจากบทเรียนในหนังสือ เพื่อเสริมความรู้และการประยุกต์ใช้ความรู้ให้กับผู้อ่าน

ตัวอย่างกิจกรรมฝึกทักษะ



กิจกรรมฝึกทักษะ

ให้นักเรียนใช้ Google Colab เขียนโปรแกรมภาษาไพทอนโดยใช้คำสั่ง print ให้ได้ผลลัพธ์ตามที่กำหนดให้ดังนี้

1. เขียนโปรแกรมแสดงผลลัพธ์ สวัสดิ์
2. เขียนโปรแกรมแสดงผลลัพธ์ Nakornpathom
3. เขียนโปรแกรมแสดงผลลัพธ์ 101
4. เขียนโปรแกรมแสดงผลลัพธ์ 5.75
5. เขียนโปรแกรมแสดงผลลัพธ์

*
**



บทสรุปท้ายบท

สรุปเนื้อหาภายในบทเรียนของหนังสือ เพื่อให้ผู้อ่านได้เรียนรู้เนื้อหาที่ได้ศึกษามาในส่วนที่สำคัญของบทเรียนที่ได้เรียนมาภายในหนังสือ

ตัวอย่างบทสรุปท้ายบท

บทสรุปท้ายบท

สรุปแล้วภาษาไพทอนเป็นภาษาที่ง่ายต่อการเรียนรู้การเขียนโปรแกรม ภาษาไพทอนพัฒนาโดย กิโด ฟาน รอสซัม (Guido van Rossum) ชื่อของภาษาไพทอน คือ ภาษาไพทอนนั้นเขียนได้ง่ายเมื่อเทียบกับภาษาคอมพิวเตอร์ส่วนใหญ่ สามารถนำไปใช้งานจริงและต่อยอดได้ง่ายซึ่งภาษาไพทอนมีไลบรารี (Library) ให้ใช้งานมากมายจึงมีผู้ใช้งานจำนวนมาก ในปัจจุบันภาษาไพทอนเป็นภาษาหนึ่งที่มีความนิยมทั่วโลก ในการเขียนโปรแกรมไพทอนจะมีเครื่องมือช่วยให้โปรแกรมทำงาน (Run) ตามคำสั่งได้ใน 2 โหมด คือ โหมดอิมมีเดียท (Immediate mode) และโหมดสคริปต์ (Script mode) เครื่องมือที่ใช้ในการเขียนโปรแกรมภาษาไพทอนยกตัวอย่าง เช่น Python IDLE, Thonny IDE, Google Colab เป็นต้น ส่วนในการแสดงผลผ่านทางหน้าจอของภาษาไพทอนจะใช้คำสั่ง print แสดงผลข้อความและตัวเลข และในการเขียนโปรแกรมเราสามารถ เพิ่มข้อความอธิบาย (Comment) ซึ่งจะช่วยอธิบายการทำงานของโค้ดในส่วนต่างๆ และอย่างไรก็ตาม ในการเขียนโปรแกรมอาจเกิดข้อผิดพลาด (Error) ขึ้นได้หากเขียนโปรแกรมไม่ถูกต้องตามหลักไวยากรณ์ของภาษา การทำความเข้าใจข้อผิดพลาดเหล่านี้จึงเป็นส่วนสำคัญในการเรียนรู้การเขียนโปรแกรมเช่นกัน



แบบฝึกหัดท้ายบท

กิจกรรมทบทวนท้ายบทเรียนของหนังสือ ในขั้นสุดท้ายของบทเรียนเพื่อทบทวนความรู้ของผู้อ่านที่ได้ศึกษามาทั้งหมดของบทเรียนภายในหนังสือ เป็นกิจกรรมที่ใช้ตรวจสอบความเข้าใจของผู้อ่าน

ตัวอย่างบทสรุปท้ายบท

แบบฝึกหัดท้ายบท

ให้ตอบคำถามและใช้ Google Colab เขียนโปรแกรมตามโจทย์ที่กำหนดให้ดังนี้

1. ให้บอกชื่อของภาษาไพทอน
2. ให้บอกชื่อเครื่องมือที่ใช้ในการเขียนโปรแกรมภาษาไพทอนมาจำนวน 5 ชื่อ
3. เขียนโปรแกรมแสดงผลลัพธ์ เป็นชื่อของตนเองภาษาอังกฤษ
4. `print("145 + 652")` ผลลัพธ์ที่ได้คืออะไร
5. เขียนโปรแกรมแสดงผลลัพธ์ต่อไปนี้ $(\wedge(oo)\wedge)$

	หน้า
คำนำ	ข
สัญลักษณ์ที่ใช้ในหนังสือ	ค
สารบัญ	ฉ
สารบัญภาพ	ช
สารบัญตาราง	ณ
บทที่ 1 รู้จักภาษาไพทอน	1
1.1 ประวัติภาษาไพทอน	2
1.2 ทำไมต้องเรียนไพทอน	4
1.3 เครื่องมือในการเขียนไพทอน	5
1.4 Thonny IDE	8
1.5 Google Colab	17
1.6 การแสดงผลในภาษาไพทอน	21
1.7 การเพิ่มข้อความอธิบาย	23
1.8 ข้อผิดพลาดในภาษาไพทอน	24
บทสรุปท้ายบท	27
แบบฝึกหัดท้ายบท	27
บทที่ 2 พื้นฐานการเขียนภาษาไพทอน	28
2.1 ตัวแปร	29
2.2 ชนิดของข้อมูลพื้นฐาน	36
2.3 การแปลงชนิดข้อมูล	37
2.4 ตัวระบุ	40
2.5 ค่าจุดลอยตัว	42
2.6 รหัสควบคุมในสตริง	44
2.7 การป้อนข้อมูลของผู้ใช้งาน	45
บทสรุปท้ายบท	49
แบบฝึกหัดท้ายบท	50
บทที่ 3 ตัวดำเนินการและนิพจน์	51
3.1 ตัวดำเนินการและนิพจน์	52
3.2 ตัวดำเนินการทางคณิตศาสตร์	52
3.3 ตัวดำเนินการเปรียบเทียบ	56
3.4 ตัวดำเนินการทางตรรกะ	58
3.5 การประยุกต์ใช้ตัวดำเนินการและนิพจน์ในการแก้ปัญหา	61
บทสรุปท้ายบท	63
แบบฝึกหัดท้ายบท	63

	หน้า
บทที่ 4 คำสั่งเงื่อนไข	64
4.1 คำสั่ง if แบบพื้นฐาน	65
4.2 คำสั่ง if-else	69
4.3 คำสั่ง if-elif-else	75
บทสรุปท้ายบท	82
แบบฝึกหัดท้ายบท	82
บทที่ 5 คำสั่งแบบวนรอบหรือทำซ้ำ	83
5.1 คำสั่ง while	85
5.2 คำสั่ง for	89
5.3 คำสั่ง Nested loop	96
บทสรุปท้ายบท	97
แบบฝึกหัดท้ายบท	97
บทที่ 6 ฟังก์ชันและลิสต์	98
6.1 การสร้างฟังก์ชัน	100
6.2 การเรียกใช้ฟังก์ชัน	102
6.3 การคืนค่าจากฟังก์ชัน	104
6.4 ขอบเขตของตัวแปร	107
6.5 การประกาศลิสต์	109
6.6 การเข้าถึงข้อมูลในลิสต์	111
6.7 การแก้ไข เพิ่ม และลบข้อมูลภายในลิสต์	112
6.8 การอ่านค่าในลิสต์ด้วย For loop	116
บทสรุปท้ายบท	118
แบบฝึกหัดท้ายบท	118
บทที่ 7 Turtle Graphic	119
7.1 ติดตั้ง ColabTurtle	121
7.2 คำสั่งที่ใช้วาดภาพ Turtle	123
7.3 การสร้างรูปเรขาคณิต	128
บทสรุปท้ายบท	134
แบบฝึกหัดท้ายบท	135
บรรณานุกรม	136
ดัชนี	137
ประวัติผู้เขียน	141
เฉลยกิจกรรมฝึกทักษะและแบบฝึกหัดท้ายบท	142

	หน้า
ภาพที่ 1.1 กีโด ฟาน รอสซัม (Guido van Rossum)	3
ภาพที่ 1.2 Python 3.x	4
ภาพที่ 1.3 ความนิยมของภาษาคอมพิวเตอร์	5
ภาพที่ 1.4 โหมดดิมมีเดียทและโหมดสคริปต์	6
ภาพที่ 1.5 Python IDLE	7
ภาพที่ 1.6 ไพทอนแบบออนไลน์ เว็บไซต์ replit.com	8
ภาพที่ 2.1 Error การแปลงเป็นตัวเลข	38
ภาพที่ 2.2 ตัวระบุในตัวแปร	41
ภาพที่ 2.3 ตัวระบุในโมดูล	42
ภาพที่ 2.4 การเปรียบเทียบค่าจุดลอยอย่างปลอดภัย	44
ภาพที่ 4.1 ผังงานการทำงานของคำสั่ง if	66
ภาพที่ 4.2 ผังงานการทำงานของคำสั่ง if-else	70
ภาพที่ 4.3 ผังงานการทำงานของคำสั่ง if-elif-else	76
ภาพที่ 5.1 การทำงานของลูป while	85
ภาพที่ 5.2 การทำงานของลูป for	90
ภาพที่ 6.1 ตัวอย่างการประกาศลิสต์	109
ภาพที่ 6.2 การอ้างอิงอินเด็กซ์ของลิสต์	111

	หน้า
ตารางที่ 2.1 ตารางแสดงการตั้งตัวระบุที่ถูกต้องและไม่ถูกต้อง	41
ตารางที่ 2.2 ตารางแสดงรหัสควบคุมที่พบบ่อย	45
ตารางที่ 3.1 ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic operators)	53
ตารางที่ 3.2 ลำดับความสำคัญของตัวดำเนินการ	55
ตารางที่ 3.3 ตัวดำเนินการเปรียบเทียบ (Comparison operators)	56
ตารางที่ 3.4 ตัวดำเนินการตรรกะ (Logical Operators)	58
ตารางที่ 3.5 ค่าความจริงของตัวดำเนินการ and	59
ตารางที่ 3.6 ค่าความจริงของตัวดำเนินการ or	59
ตารางที่ 3.7 ค่าความจริงของตัวดำเนินการ not	59
ตารางที่ 7.1 กลุ่มคำสั่งการเคลื่อนที่	123
ตารางที่ 7.2 กลุ่มคำสั่งการควบคุม	126

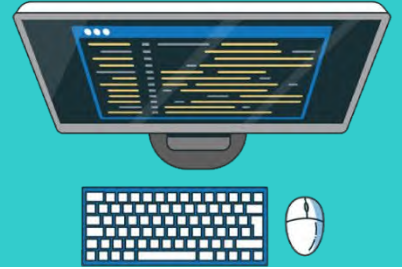


บทที่ 1 รู้จักภาษาไพทอน



จุดประสงค์การเรียนรู้

1. มีความรู้ ความเข้าใจภาษาไพทอน
2. สามารถบอกถึงเครื่องมือในการเขียนภาษาไพทอน
3. สามารถเขียนโปรแกรมแสดงผลในภาษาไพทอนได้



ภาษาคอมพิวเตอร์ (Computer Language) คือ ภาษาที่ผู้ใช้งานใช้สื่อสารกับคอมพิวเตอร์ หรือคอมพิวเตอร์ใช้สื่อสารกัน แล้วทำให้คอมพิวเตอร์สามารถทำงานได้ตามคำสั่งนั้นได้ ซึ่งมักใช้เรียกแทนภาษาโปรแกรม แต่ความเป็นจริงแล้วภาษาโปรแกรมคือส่วนหนึ่งของภาษาคอมพิวเตอร์เท่านั้น โปรแกรมภาษาที่นิยมใช้ในการเขียน เช่น ภาษาซี (C Language) ภาษาโคบอล (Cobol) ภาษาซีพลัส (C++) ภาษาจาวา (Java) และภาษาไพทอน (Python) เป็นต้น ซึ่งในแต่ละภาษาจะมีวิธีการเขียนที่คล้ายกันแต่มีความซับซ้อนที่แตกต่างกัน ภาษาไพทอน (Python) เป็นภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่งที่มีความสามารถสูง ดังนั้นภาษาไพทอนจึงเป็นภาษาที่อยู่ในระดับสูง โดยถูกออกแบบมาให้เป็นภาษาสคริปต์ที่อ่านง่าย ภาษาไพทอนนั้นตัดความซับซ้อนของโครงสร้างและไวยากรณ์ของภาษาออกไป ในส่วนของการแปลงชุดคำสั่งที่เขียนให้เป็นภาษาเครื่องไพทอน มีการทำงานแบบอินเทอร์พรีเตอร์ (Interpreter) คือเป็นการแปลชุดคำสั่งทีละบรรทัด นอกจากนี้ยังสามารถนำไปใช้ในการเขียนโปรแกรมได้หลากหลายประเภท ในบทนี้จะนำเสนอเนื้อหาเกี่ยวกับประวัติความเป็นมาของภาษาไพทอน ทำไมถึงต้องเรียนภาษาไพทอน เครื่องมือในการเขียนไพทอนและ

1.1 ประวัติภาษาไพทอน

ภาษาไพทอนเป็นภาษาที่ง่ายต่อการเริ่มต้นเรียนรู้การเขียนโปรแกรม มีโครงสร้างการทำงานที่ไม่ซับซ้อน มีชุดคำสั่งที่ทำงานทางด้านกราฟิกให้เลือกให้ใช้งานได้สะดวก สามารถทดสอบการทำงานตามคำสั่ง และตรวจสอบผลลัพธ์ได้ทันที เนื่องจากภาษา ไพทอนเป็นภาษาที่เรียนรู้ได้ง่าย รวดเร็ว มีประสิทธิภาพสูง และรองรับงานด้านต่างๆ เช่น งานวิทยาศาสตร์ วิศวกรรมศาสตร์ และวิทยาศาสตร์ข้อมูล จึงเป็นภาษายอดนิยมสำหรับการใช้ในการเขียนโปรแกรม

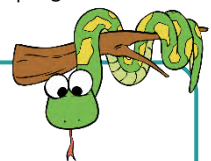


ที่มาภาพ: https://commons.wikimedia.org/wiki/File:Logo_python.png



เกร็ดความรู้

Python อ่านว่า ไพทอน ซึ่งในภาษาอังกฤษหากเราแปลเป็นไทยคือ “งูหลาม”



สำหรับประวัติภาษาไพทอนนั้นกำเนิดขึ้นในปลายปี ค.ศ. 1980 และการพัฒนาได้เริ่มต้นขึ้นในเดือน ธันวาคม ปีค.ศ. 1989 โดย กีโด ฟาน รอสซัม (Guido van Rossum) ที่ Centrum Wiskunde & Informatica (CWI) ซึ่งเป็นสถาบันวิจัยทางด้านคณิตศาสตร์และวิทยาการคอมพิวเตอร์ในเมืองอัมสเตอร์ดัม ประเทศ เนเธอร์แลนด์ ในเวลานั้น กีโด ฟาน รอสซัม ต้องพัฒนาโปรแกรมสำหรับผู้ดูแลระบบ เพื่อใช้ในโครงการ Amoeba ซึ่งเป็นโครงการเกี่ยวกับระบบปฏิบัติการแบบกระจาย (Distributed Operating System) อย่างไรก็ตามเขารู้สึกว่าภาษาโปรแกรม ABC, C และ Bourne Shell มีข้อจำกัดมากมาย ทั้งเรื่องใช้เวลาในการพัฒนานานมาก และไม่สามารถตอบโจทย์หลายประการ ดังนั้น กีโด ฟาน รอสซัม จึงได้ตัดสินใจเริ่มพัฒนา ภาษาโปรแกรมระดับสูงขึ้นมาใหม่เพื่อใช้งานเองเป็นงานอดิเรก โดยนำเอาสิ่งที่ชอบในภาษา ABC มาพัฒนาลง ไปในภาษาโปรแกรมไพทอน รวมถึงได้พัฒนาส่วนอื่นๆ เพิ่มเติมเข้าไป และในเวลาต่อมาจึงได้เผยแพร่ Python 1.0 เวอร์ชันแรกในปี ค.ศ. 1994 หากเทียบกับภาษา Java ที่ได้ทำการเผยแพร่เวอร์ชันแรก ในปี ค.ศ. 1996 จะเห็นได้ว่าภาษาไพทอนมีอายุนานกว่าภาษา Java ถึง 2 ปี



ภาพที่ 1.1 กีโด ฟาน รอสซัม (Guido van Rossum)

ที่มาภาพ: <https://commons.wikimedia.org/wiki/File:Guido-portrait-2014-drc.jpg>

สำหรับที่มาของชื่อภาษาโปรแกรมไพทอนนั้นไม่ได้มีที่มาเกี่ยวข้องกับ “งู” เหมือนกับชื่อของมันแต่อย่างใด แต่ในช่วงที่ตัดสินใจเลือกใช้นั้น ชื่อแรกที่เข้ามาในความคิดของ กีโด ฟาน รอสซัม ก็คือ “มอนตี ไพทอน: ละครสัตว์เหินหาว” (Monty Python’s Flying Circus) ซึ่งเป็นชื่อรายการโทรทัศน์ทางช่อง BBC แนวตลกชื่อดังจากฝั่งอังกฤษที่เขาชื่นชอบมาก โดย กีโด ฟาน รอสซัม ให้เหตุผลว่า “Python” เป็นชื่อที่สั้น จำได้ง่าย ฉีกแนวนิดๆ และดูลึกลับ ในตอนนั้นโดยทั่วไปมักจะนิยมเอาชื่อของบุคคลที่มีชื่อเสียงมาใช้เป็นชื่อ ภาษาโปรแกรมคอมพิวเตอร์ เช่น Ada, Pascal และ Eiffel เป็นต้น ถึงแม้ว่าที่มันักแสดงในรายการจะไม่ได้มีชื่อเสียงทางด้านวิทยาศาสตร์และเทคโนโลยี แต่ก็เป็นที่ชื่นชอบในกลุ่มชาว Geek เป็นอย่างมาก รวมถึงกลุ่ม คนที่ทำงานใน CWI ก็มักจะนิยมเอาชื่อรายการทีวีโชว์มาตั้งชื่อในงานของตัวเองอีกด้วย นี่จึงเป็นเหตุผลที่ มาที่ไปของชื่อภาษาไพทอน นอกจากนั้น กีโด ฟาน รอสซัม ยังใช้ชื่อของนักแสดงตลกชาวอังกฤษชื่อดังและเป็น หนึ่งในสมาชิกผู้ก่อตั้งทีม Monty Python ที่ชื่อ Eric Idle มาใช้เป็นชื่อ IDE หรือเครื่องมือที่ใช้ในการพัฒนา โปรแกรมว่า “IDLE” อีกด้วย

FUNCTION



LIST

0 1 2
[A, B, C]



บทที่ 6 ฟังก์ชันและลิสต์

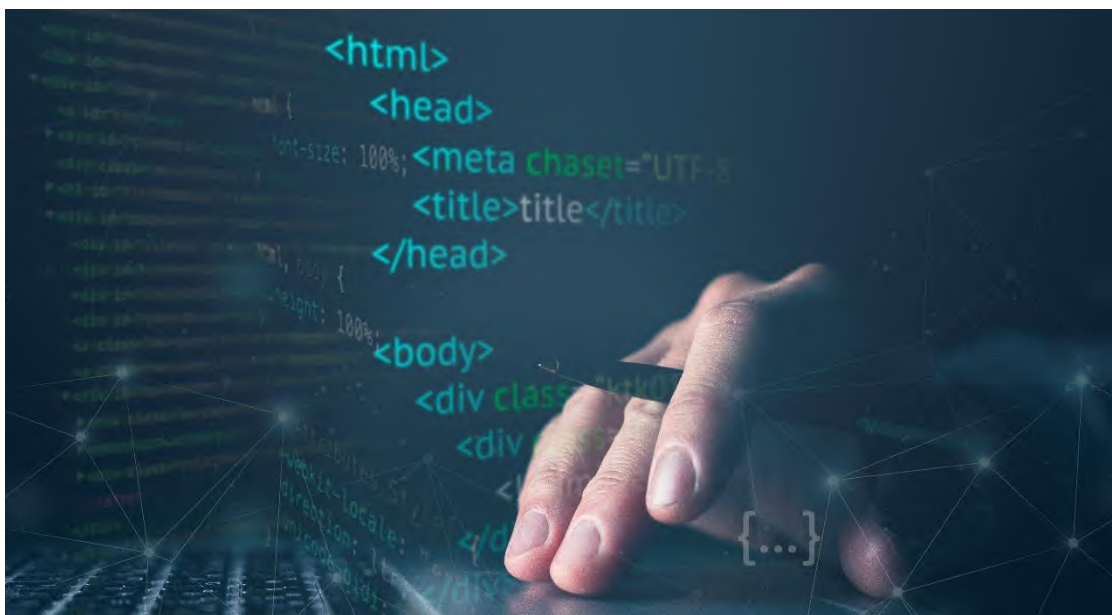


จุดประสงค์การเรียนรู้

1. อธิบายความสำคัญและประโยชน์ของการใช้ฟังก์ชันได้
2. สร้างและเรียกใช้ฟังก์ชันแบบมีและไม่มีพารามิเตอร์ได้
3. อธิบายความหมายและประโยชน์ของโครงสร้างข้อมูลแบบลิสต์ได้
4. สร้าง เข้าถึง และแก้ไขข้อมูลในลิสต์ได้อย่างถูกต้อง



ในบทก่อนๆ เราได้เรียนรู้วิธีการเขียนโปรแกรมคำสั่งแบบวนรอบหรือทำซ้ำ แต่เมื่อโปรแกรมมีความซับซ้อนมากขึ้น การเขียนโปรแกรมทุกอย่างรวมกันจะทำให้จัดการได้ยากและเกิดความซ้ำซ้อน ในบทนี้เราจะมาเรียนรู้เครื่องมือสำคัญ 2 อย่างที่จะช่วยให้โปรแกรมของเรามีระเบียบมากยิ่งขึ้น นั่นคือ ฟังก์ชัน (Function) สำหรับจัดกลุ่มคำสั่งที่ทำงานเฉพาะอย่าง และ ลิสต์ (List) สำหรับรวบรวมและจัดการข้อมูลหลายๆ ค่าไว้ในที่เดียว เหมือนกับการจัดเครื่องมือใส่กล่องและการจัดของใส่ชั้นวางเพื่อให้หยิบใช้งานง่ายขึ้นเอง



ฟังก์ชัน เปรียบเสมือนการสร้างเครื่องมือเฉพาะทางของคุณเองขึ้นมา คุณออกแบบและสร้างมันเพียงครั้งเดียวแล้วเก็บมันไว้ในกล่องเครื่องมือของคุณ เมื่อไหร่ก็ตามที่ต้องการใช้งานก็แค่หยิบมันออกมาเรียกใช้โดยไม่ต้องสร้างใหม่ทั้งหมด ส่วนลิสต์ เปรียบเสมือนการมีตู้เก็บเอกสารที่มีลิ้นชักเรียงกันเป็นลำดับหมายเลข คุณสามารถตั้งชื่อตู้เอกสารนี้ได้เพียงชื่อเดียว และนำเอกสารหรือข้อมูลแต่ละชิ้นไปเก็บในลิ้นชักแต่ละใบได้อย่างเป็นระเบียบ ทำให้คุณสามารถเข้าถึง แก้ไข เพิ่ม หรือลบข้อมูลเป็นกลุ่มได้อย่างง่ายดาย



เกร็ดความรู้

ฟังก์ชัน (Function) คือ กลุ่มของโปรแกรมที่ถูกกำหนดขึ้นโดยมีชื่อเรียก เราใช้ฟังก์ชันเมื่อต้องการทำงานอย่างเดียวกันซ้ำๆ หลายครั้งโดยไม่ต้องเขียนโปรแกรมเดิมอีก ฟังก์ชันสามารถรับค่าที่ส่งเข้ามา (Arguments) และส่งค่ากลับ (Returns the value) ได้



6.1 การสร้างฟังก์ชัน (Creating Functions)

การเขียนโปรแกรมโดยแยกการทำงานแต่ละส่วนออกชัดเจนเป็นฟังก์ชันมีข้อดีคือ ทำให้การเขียนโปรแกรมเป็นไปด้วยความรวดเร็ว เพราะสามารถแยกกันเขียนโปรแกรมแต่ละส่วนแล้วนำมารวมกันได้ อีกทั้งเมื่อเกิดปัญหาก็สามารถหาจุดผิดพลาดของโปรแกรมได้ง่ายและสะดวกกว่าการเขียนโปรแกรมรวมกันในโปรแกรมหลัก

ฟังก์ชัน (Function) หรือโปรแกรมย่อย (Subroutine) เป็นกลุ่มคำสั่งที่มีการนิยามชื่อที่ใช้สำหรับอ้างอิงแทนกลุ่มคำสั่งเหล่านั้น เพื่อให้สามารถเรียกทำงานได้ตามต้องการ กลุ่มของคำสั่งที่ประกอบกันขึ้นเป็นฟังก์ชันจะทำหน้าที่อย่างใดอย่างหนึ่งโดยเฉพาะ หลักการสำคัญฟังก์ชันช่วยให้เราเขียนโปรแกรมครั้งเดียว แต่เรียกใช้ได้หลายครั้ง (Reusability) โครงสร้างพื้นฐานของฟังก์ชันในภาษาไพทอนจะขึ้นต้นด้วยคีย์เวิร์ด **def** (ย่อมาจาก define) ตามด้วยชื่อฟังก์ชัน วงเล็บ () และปิดท้ายด้วยเครื่องหมายโคลอน

```
def function_name(parameter1, parameter2):
    # บล็อกของโค้ดที่ต้องการให้ฟังก์ชันทำงาน (ต้องย่อหน้า)
    # ...
    return result # การส่งค่ากลับ (ถ้ามี)
```

โดย

def:	คือ	คำสั่งเริ่มต้นการสร้างฟังก์ชัน
function_name	คือ	ชื่อฟังก์ชันที่เราตั้งเอง ควรเป็นชื่อที่สื่อความหมาย
parameter	คือ	ตัวแปรที่รองรับค่าจากภายนอก ที่เราส่งให้ฟังก์ชันนำไปประมวลผล (อาจจะมีหรือไม่มีก็ได้)
return	คือ	คำสั่งสำหรับส่งค่าผลลัพธ์กลับออกไปจากฟังก์ชัน (อาจจะมีหรือไม่มีก็ได้)

ในรูปแบบของการสร้างฟังก์ชันในภาษาไพทอนนั้น จะใช้คำสั่ง **def** และหลังจากนั้น **function_name** เป็นชื่อของฟังก์ชันและในวงเล็บ () เป็นการกำหนดพารามิเตอร์ของฟังก์ชัน พารามิเตอร์ของฟังก์ชันนั้นสามารถมีจำนวนเท่าไรก็ได้หรือไม่มีก็ได้ และเช่นเดียวกับภาษาอื่นๆ ฟังก์ชันอาจจะมีหรือไม่มี การส่งค่ากลับ สำหรับฟังก์ชันที่ไม่มีการ **return** ค่ากลับนั้น เรามักจะเรียกว่า **โพรซีเจอร์ (Procedure)** ต่อไปมาดูตัวอย่างการสร้างฟังก์ชันในภาษาไพทอน



PYTHON TURTLE GRAPHICS

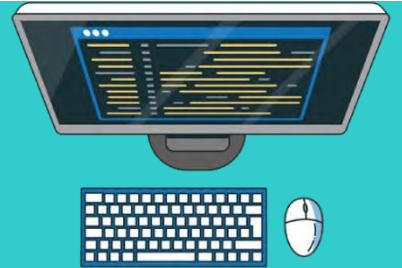


บทที่ 7 Turtle Graphic



จุดประสงค์การเรียนรู้

1. มีความรู้และเข้าใจการเขียนโปรแกรมกราฟิก โดยใช้ภาษาไพทอน
2. สามารถเขียนโปรแกรมโดยใช้คำสั่ง Turtle of Python
3. สามารถประยุกต์ใช้โปรแกรมกับรูปเรขาคณิตได้



ในปัจจุบันมีการใช้งานกราฟิกในทุกๆ ด้านไม่ว่าจะเป็นด้านธุรกิจ ด้านงานพิมพ์ ด้านงานศิลปะ ด้านการบันเทิง ด้านงานโฆษณา และด้านการศึกษา ซึ่งเห็นได้ชัดว่างานกราฟิกนั้นมีบทบาทสำคัญมากในปัจจุบัน ในการเขียนโปรแกรมก็มีการสร้างงานด้านกราฟิก โดยภาษาไพทอนจะมีการสร้างกราฟิกด้วยเต่าไพทอน (Python Turtle) ในบทนี้จะนำเสนอเกี่ยวกับไลบรารี (Library) เสริมที่ติดตั้งมาพร้อมกับไพทอน คำสั่งต่างๆ และการประยุกต์ใช้งานในการสร้างรูปเรขาคณิตต่างๆ



เต่าไพทอนนั้นมีเพื่อส่งเสริมให้ผู้เขียนโปรแกรมหรือผู้ที่ต้องการสร้างกราฟิกสามารถสร้างได้ง่ายขึ้น โดยไม่จำเป็นต้องมีพื้นฐานมาก่อน ด้วยการเขียนโปรแกรมแบบโลโก้ คล้ายกับการเขียนภาพบนกระดานดำนั่นเอง ในการเดินหรือการวาดรูปเต่าไพทอนจะมีคำสั่งต่างๆ ที่ใช้ในการเคลื่อนที่ โดยในการเริ่มต้นของการให้เต่าวาดรูป ซึ่งต้องใช้คำสั่ง `Import Turtle` มาจาก Library ก่อนถึงจะสั่งให้เต่าวาดรูปได้



เกร็ดความรู้

ไลบรารี (Library) เป็นที่เก็บฟังก์ชันการทำงานสำเร็จรูปไว้เพื่อเรียกใช้งาน โดยไม่จำเป็นต้องเขียนคำสั่งใหม่



7.1 ติดตั้ง ColabTurtle

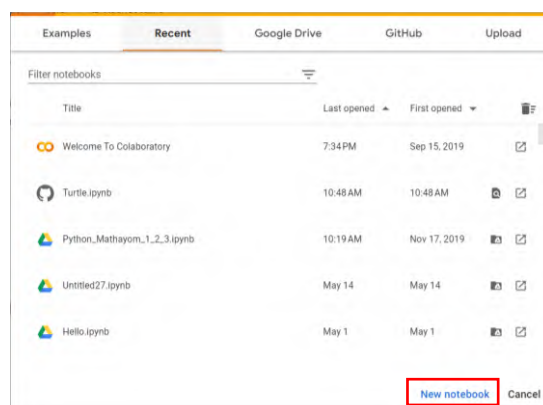


การเขียนโปรแกรมภาษาไพทอน ด้วย Google Colab จำเป็นต้องติดตั้ง ColabTurtle ก่อนถึงจะสามารถใช้งานกราฟิกในส่วนของเราได้ โดยในส่วนหัวของโปรแกรมต้องมีข้อความ 2 บรรทัดนี้ และทำการรันโปรแกรมก่อน เริ่มการเขียนโปรแกรมตัวไพทอน

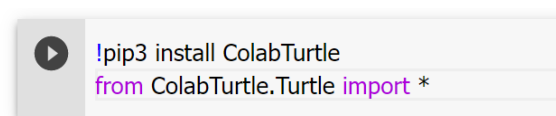
```
!pip3 install ColabTurtle
from ColabTurtle.Turtle import *
```

ซึ่งวิธีการติดตั้ง ColabTurtle ใน Google Colab มีดังนี้

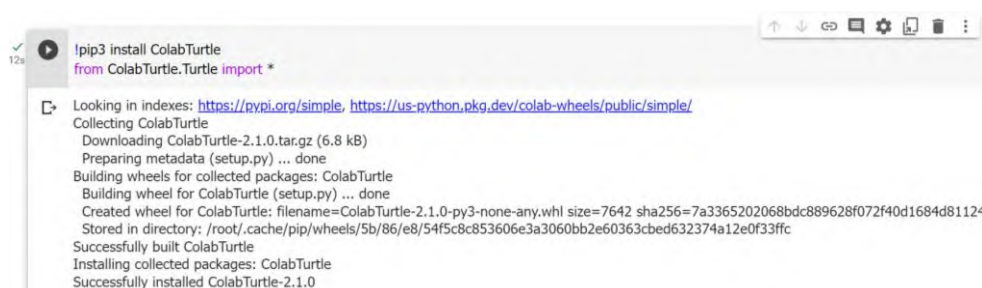
1. เข้าเว็บ Google Colab ที่ <https://colab.research.google.com>
2. คลิกที่ “New notebook”



3. พิมพ์คำสั่งการติดตั้ง ColabTurtle



4. คลิกที่ปุ่มรัน (Run) หรือกดปุ่มคีย์ลัด “Ctrl + Enter”





python

เรียนรู้ภาษาไพทอนด้วยหนังสือที่เข้าใจง่าย สำหรับผู้เริ่มต้นสู่โลก
การเขียนโปรแกรมก้าวสู่โลกของการเขียนโปรแกรมด้วยภาษาไพทอน
ภาษาโปรแกรมที่กำลังได้รับความนิยมสูงสุดในปัจจุบัน ด้วยเนื้อหาที่
เรียบเรียงอย่างเป็นระบบ เข้าใจง่าย เหมาะสำหรับผู้เริ่มต้นที่ไม่มีพื้นฐาน
การเขียนโปรแกรมมาก่อน

หนังสือเล่มนี้จะพาคุณเริ่มต้นจากพื้นฐานสู่การประยุกต์ใช้งาน
จริง ผ่านการอธิบายที่ละเอียด ตัวอย่างโค้ดที่ชัดเจน พร้อมแบบฝึกหัด
ให้ได้ลงมือปฏิบัติจริง ครบถ้วนเนื้อหาสำคัญของไพทอน

ISBN 978-616-629-320-3



9 786166 293203

ราคา 149 บาท